IN THE CLAIMS

1. (Currently Amended) A device for combining a plurality of arithmetic flags, comprising:

a combination function module that examines a plurality of arithmetic flags, determines field size of the plurality of arithmetic flags and based on the determination of the field size will logically combine the plurality of arithmetic flags into a single combined arithmetic flag variable, wherein the plurality of arithmetic flags represent [[the]] a status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items.

2. (Original) The device recited in claim 1, further comprising:

a condition check module that determines the status of the combined arithmetic flag variable and causes the processor to execute an appropriate operation based on the status.

3. (Previously Presented) The device recited in claim 1, wherein the field size is based on either a nibble, byte, half word, or word in length.

4. (Original) The device recited in claim 3, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

5. (Original) The device recited in claim 4, the combination function module performs either an AND or an OR operation.

6. (Previously Presented) The device recited in claim 2, wherein the status determined by the condition check module further comprises:

any data item has overflowed;

any data item has not overflowed;

any data item is positive or zero;

any data item is negative;

any data item is zero;

any data item is not zero;

any data item has a carry out;

any data item does not have a carry out;

all data items have overflowed;

all data items have not overflowed;

all data items are positive or zero;

all data items are negative;

all data items are zero;

all data items are not zero;

all data items have a carry out; and

all data items do not have a carry out.

7. (Currently Amended) A method of combining a plurality of arithmetic flags for presentation to a processor, comprising:

determining a field size of the plurality of arithmetic flags on which to base a combination process, wherein the plurality of arithmetic flags represent [[the]] a status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items;

extracting the plurality of arithmetic flags based on the field size;

logically combining the plurality of arithmetic flags based on a function selected when a combination process is selected; and

storing a result of the combining of the plurality of arithmetic flags in a destination register for access by the processor.

8. (Original) The method recited in claim 7, wherein the field size is based either a nibble, byte, half word, or word in length.

9. (Original) The method recited in claim 8, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

10. (Original) The method recited in claim 9, wherein the function further comprises:
an AND or OR operation.

11. (Original) The method recited in claim 10, wherein the function may be used to determine the status of the plurality of data items, said status comprising:

any data item has overflowed;

any data item has not overflowed;

any data item is positive or zero;

any data item is negative;

any data item is zero;

any data item is not zero;

any data item has a carry out;

any data item does not have a carry out;

all data items have overflowed;

all data items have not overflowed;

all data items are positive or zero;

all data items are negative;

all data items are zero;

all data items are not zero;

all data items have a carry out; and

all data items do not have a carry out.

12. (Currently Amended) An apparatus comprising a data storage medium for storing instructions, the instructions when executed by a processor result in the processor performing a method, the method comprising:

determining a field size of the plurality of arithmetic flags on which to base a combination process, wherein the plurality of arithmetic flags represent [[the]] a status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items;

extracting the plurality of arithmetic flags based on the field size;

logically combining the plurality of arithmetic flags based on a function selected when a combination process is selected; and

storing a result of the combining of the plurality of arithmetic flags in a destination register for access by the processor.

13. (Previously Presented) The apparatus recited in claim 12, wherein the field size is based on either a nibble, byte, half word, or word in length.

14. (Original) The apparatus recited in claim 13, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

15. (Original) The apparatus recited in claim 14, wherein the function further comprises an AND or OR operation.

16. (Original) The apparatus recited in claim 15, wherein the function may be used to determine the status of the plurality of data items, said status comprising:

any data item has overflowed;

any data item has not overflowed;

any data item is positive or zero;

any data item is negative;

any data item is zero;

any data item is not zero;

any data item has a carry out;

any data item does not have a carry out;

all data items have overflowed;

all data items have not overflowed;

all data items are positive or zero;

all data items are negative;

all data items are zero;

all data items are not zero;

all data items have a carry out; and

all data items do not have a carry out.

17. (Currently Amended) A method of extracting a plurality of arithmetic flags for presentation to a processor, comprising:

determining a field size of the plurality of arithmetic flags on which to base a combination process, wherein the plurality of arithmetic flags represent [[the]] a status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items;

extracting the plurality of arithmetic flags based on the field size, the plurality of arithmetic flags associated with a single selected data item; and

storing a result of the extracting of the plurality of arithmetic flags in a destination register for access by the processor.

18. (Original) The method recited in claim 17, wherein the field size is based either a nibble, byte, or half word in length.

19. (Original) The method recited in claim 18, wherein the plurality of arithmetic flags further comprise:

a negative data value, a zero data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items.

20-22. (Canceled)

23. (New) A system for combining a plurality of arithmetic flags, comprising:

a combination function module that examines a plurality of arithmetic flags, determines field size of the plurality of arithmetic flags and based on the determination of the field size will logically combine the plurality of arithmetic flags into a single combined arithmetic flag variable, wherein the plurality of arithmetic flags represent a status of a plurality of data items after a mathematical operation is performed on the plurality of data items,

and a processor including a condition check module coupled to the combination function module, the processor to receive the single combined arithmetic flag variable and to determine the next operation to perform based upon the status of the single combined arithmetic flag variable.

24. (New) The system of claim 23, wherein the processor includes at least three stages of pipelining.

25. (New) The system of claim 23, wherein the at least three stages of pipelining includes a fetch stage, a decode stage, and an execute stage.